UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/766,209	01/19/2001	Brandon J. Passanisi	P5505/14695.007001	9219

32615 7590 08/12/2005

OSHA LIANG L.L.P./SUN
1221 MCKINNEY, SUITE 2800
HOUSTON, TX 77010

EXAMINER	
VU, TUAN A	

ART UNIT	PAPER NUMBER
2193	

DATE MAILED: 08/12/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/766,209	PASSANISI, BRANDON J.	
	Examiner	Art Unit	
	Tuan A. Vu	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *01 June 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3-6,8-11,13-18 and 20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3-6,8-11,13-18 and 20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 6/1/2005.

As indicated in Applicant's response, claims 1, 3-4, 11, 18 have been amended, with claims 2, 7, 12 and 19 canceled. Claims 1,3-6,8-11,13-18, and 20 are pending in the office action.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1, 3-6,8-11, 13-18, and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kobayashi, USPN: 6,633,888 (hereinafter Kobayashi) in view of admitted prior art (see Affidavit 1.132 filed 9/24/2004 and Anne Thomas' White paper -- hereinafter APA).

As per claim 1, Kobayashi discloses an apparatus for facilitating development of Java bundles, comprising:

a processor and a memory;

an integrated Development Environment to execute a module under the control of the processor to generate Java bundles (e.g. *visual builder 708* - Fig. 7; *JAR file* - step *808*, Fig. 8; step *1910*, Fig. 19; col. 7, lines 45-55 – Note: *Visual builder 708* reads on computer executing module for bean add-ons development purpose) using a plurality of development tools (*IDE 712* – Fig. 7; col. 11, lines 26-30 – Note: a IDE reads on plurality of tools), wherein the tools

comprise a manifest generator tool configured to create a manifest file for the bundles (e.g. creator 308 – Fig. 3).

But Kobayashi does not disclose that the Java bundles are Java Embedded Server bundles nor the manifest file being JES manifest file. The use of Java compacted packages like JAR in network distribution to facilitate their deployment at target devices where storage resources are limited, e.g. resources-restraint embedded devices where was a known concept at the time the invention was made. This concept is evidenced via embedded systems with capabilities to install and deploy portable package of software such as the likes Java Embedded Servers as taught by APA (see Affidavit 1.132 filed 9/24/2004 and Anne Thomas' White paper). Based on Kobayashi's suggestion that cross-platform Java packages (or JAR files) or bytecodes can be deployed on embedded processing units (e.g. step *808*, Fig. 8; col. 7, lines 19-30) it would have been obvious for one of ordinary skill in the art at the time the invention was made to implement the Java package builder method by Kobayashi so that the Java bundles along with their manifest file are created using a IDE as mentioned above for facilitating the user-driven and dynamic creation and deployment at embedded systems in general and in JES in particular as by APA because, by bundling Java deployable components in small devices (or dedicated servers like JES) whose resources can be a limiting factor, the embedded systems (e.g. JES as one intended use) storage resources for otherwise non-bundled components would be alleviated, and also because according to APA, this IDE building application would support dynamic and as-needed application deployment on embedded systems utilizing, among other benefits, the known transportability of Java bytecodes (as mentioned by Kobayashi col. 7, line 46 to col. 8, line 7),

whose implicit cross-platform portability can improve deployment of program components over the network as shown by JES/APA.

As per claim 3, Kobayashi discloses a window menu within a visual builder to retrieve components to assemble a Java bundle in the IDE (e.g. Fig. 7; Fig. 13-17; step *1910* – Fig. 19; col. 11, lines 26-30); hence has disclosed implicitly disclosed drop-down menu for accessing the module as to create Java bundles.

As per claim 4, Kobayashi discloses a update mechanism in the IDE (e.g. EDIT – Fig. 11; Fig. 13-17; step *2014* – Fig. 20).

As per claim 5, Kobayashi discloses a code template (e.g. Fig. 4; Constructor bean 1000, method bean 1012 – Fig. 10 – Note: a container for Java code methods and attributes is equivalent to class templates containing elements from which to construct further code classes)

As per claim 6, Kobayashi discloses a interface template (e.g. *palette ... internal interface* - col. 17, line 62 to col. 18, 8; col. 11, lines 45-49; *environment add-on 700*– Fig. 7; Fig. 14, 17); and implementation template (e.g. *visual builder 214* – Fig. 2; transport *API* 206 – Fig. 2; Fig. 9—Note: using a visual palette to effect APIs and interface calls to constructs classes and methods for beans reads on implementation template) but only teaches a activator functionality based on events (e.g. Fig. 16, 18-20 – Note: visual display to applying binding and linking properties of created components and to test beans reads on activator template). Given the visual aspect of activating the created components based on the builder template thus suggested by Kobayashi, it would have been obvious for one of ordinary skill in the art at the time the invention was made to add to the visual menu-driven tool of Kobayashi a activator template with which the components can be activated or linked just as suggested above because

providing all the event-based functionalities for test or dynamic binding in one such graphical container module, i.e. a template, would provide the integration tool with one differentiated graphical module encompassing all the debugging and testing functions typical of the post-implementation stages of development as suggested above prior to delivering the package as built.

As per claim 8, Kobayashi (in view of APA) discloses a Java Embedded Server (JES) jar packager tool that packages JES bundles (e.g. step 1910 – Fig. 19).

As per claim 9, Kobayashi does not disclose link to JES-related web pages but teaches a Java bean-compliant visual tool of the likes of Visual Age *WebRunner* for a world wide web with Corba connectivity (e.g. *CORBA* - col. 5, lines 23-31; col. 11, lines 8-17; col. 23, lines 22-34). The creation of links to remote network storages is implied or explicit in all visual development tool (e.g. Kobayashi: Fig. 3-9 – Note: for example, CORBA and RMI retrieval of remote beans/reuse component in a Web-based connection is implied that web page link, e.g. URL, are implicit) or in JES/APA when web and browser application-based tools (with inherent web protocol connectivity features) are taught (see APA White Paper). Hence, it would have been obvious for one of ordinary skill in the art at the time the invention was made to create the visual builder by Kobayashi so that using a JES as mentioned in claim 1, JES-related links to remote storage or URL-directed pages are effected via browser pages are effected in the builder because these would enable retrieval of components supporting the IDE development at the JES as taught by APA and initially desired by the bean-compliant visual builder by Kobayashi.

As per claim 10, this claim recites code template, manifest generator tool, and jar packager tool. All of which limitations have been addressed in claims, 5, 7, and 8 respectively.

As per claim 11, Kobayashi (in combination with APA) discloses a method comprising a processor, a memory as well as software instructions stored therein as recited in the apparatus claim 1, including combining a plurality of development tools in a module for the creation of Java bundles and execute the module in an IDE (e.g. *visual builder 708* - Fig. 7; Fig. 8; step *1910*, Fig. 19; col. 7, lines 45-55 – Note: *Visual builder 708* reads on module for bean add-ons and development modules being integrated in the IDE environment , *IDE 712*) wherein the tools comprise a manifest generator tool configured to create a manifest file for the bundles (e.g. creator 308 – Fig. 3).

But Kobayashi does not specifically disclose that the bundles are Java Embedded Server bundles and that the manifest files are JES manifest files. This limitation has been however addressed in claim 1 above.

As per claim 13, Kobayashi discloses code samples (refer to rejection of claim 5).

As per claims 14 and 15, these claims correspond to the limitations of claims 7 and 8, hence are rejected with the corresponding rejection as set forth therein respectively.

As per claim 16, refer to rejection of claim 9.

As per claim 17, refer to claim 10.

As per claim 18, Kobayashi discloses an apparatus for facilitating development of Java bundles, comprising means for:

providing sample code segments (refer to rejection of claim 5);

creating Java manifest files for the bundles (re to rejection of claim 1); and

packaging the bundles (re claim 8);

executing a module into an Integrated Development Environment (e.g. *IDE 712* – Fig. 7; col. 11, lines 26-30), the module comprising a plurality of development tools (e.g. Fig. 7 - Note: *Visual builder 708* reads on module for bean add-ons and development modules being integrated in the IDE environment , *IDE 712*); wherein the tools comprise a manifest generator tool configured to create a manifest file for the bundles (e.g. creator 308 – Fig. 3).

But Kobayashi does not specifically disclose that the Java bundles are Java Embedded Server bundles and that the manifest files are JES manifest files. These limitations have been however addressed in claim 1 above.

As per claim 20, Kobayashi discloses an apparatus for facilitating development of Java bundles, comprising means for

combining, in a module, a plurality of development tools used in the creation of Java bundles (refer to corresponding rejection set forth in claim 1); and

integrating the module into a IDE (*IDE 712* – Fig. 7; col. 11, lines 26-30– Note: the use of IDE inherently teaches integration of all the products generated by the module into the development environment for creating the bundles).

But does not disclose that the Java bundles are JES bundles. However, this limitation has been addressed in claim 1 from above.

Response to Arguments

4. Applicant's arguments filed 6/1/2005 have been fully considered but they are not persuasive. Following are the observations in regard thereto.

(A) Applicant has submitted that neither Koyabashi nor APA discloses a manifest generator tool in an IDE for generation of JES bundles and that one skill in the art would not find including

a manifest generator tool in a IDE to be obvious (Appl. Rmrks, pg. 7, 2nd para). In response, it is noted that an Integrated Development Environment is defined as a 'set of integrated tools for developing software, run from one user interface and consist of a compiler, an editor, and a debugger, among others' (see Definition of Microsoft Computer Dictionary, 5th edition). The VEA's tool of Koyabashi combines a plurality of tools into a visual add-on tool wherein the IDE as well as a bean compiler are also disclosed to be part of such builder environment (see col. 8, lines 20-65; col. 11, lines 7-30). The IDE and the compiler means - to bundle beans and to include the manifest creator -- are integral to the overall visual builder, making what amounts to an Integrated Development Environment to read exactly on what Koyabashi's visual builder amounts to in view of the above definition. The claims 1 and claim 11 recite 2 limitations, respectively, 'wherein the plurality of development tools comprise a ... generator tool' (re claim 1), and 'combining in a module a plurality of development tools ... execute the module in an Integrated Development Environment' (re claim 11). Koyabashi has disclosed in light of the beans compiler of Fig. 3, a manifest generator among a plurality of development tools, such plurality of development tools being combined into the visual builder executing with a IDE and a compiler such as Koyabashi discloses via the VEA of Fig. 7; hence the visual builder module is being treated here as a software entity like a visual tool or module comprising more sub-modules, e.g. a IDE or a compiler, and performing a development task in synergy; all this in combination read on an integrated development environment definition. The argument about the advantage of having an IDE that allows a manifest generator to execute would have been moot because it did not address directly the features being explicitly claimed; hence amounts to mere allegations likely garnered from the specifications not from the claim.

(B) Applicant has submitted that the IDE of Kobayashi is not used to generate any type of bundles using a manifest generator that allows ... information for each manifest file' (Appl. Rmrks, pg. 7, 3rd para). Again, the claims only recite (i) 'an Integrated Development Environment ... to execute a module ... wherein ... the plurality of development tools comprise ... manifest generator tool' (re claim 1); or (ii) 'combine, in a module, a plurality of ... tools used ... creation of ...bundles; and execute the module in an ... (IDE), wherein ... plurality of ... tools comprise ... generator tool ..." (re claim 11). Koyabashi has disclosed a global module being a visual builder (VEA), such builder including all the components which amount to what is called an IDE from the definition set forth above, e.g. a visual editor for assembly software beans, a compiler including a generator of manifest when assembling packages; i.e. a integrated tool overall to be executed in combination (see Fig. 3-7). Such module is being executed to create bundles; such module comprises a plurality of tools among which a manifest generator is included; hence, such module amounts to an integrated development environment and it is being executed. The fact that such builder includes an IDE (Fig. 3) does not negate the fact that it by itself reads on the definition as set forth in section A and consequently on the limitation above. The claims fail to show in addition to having an IDE executing a module how more specific this execution is when for example matched with Koyabashi's visual builder which clearly represents an IDE capability and which is also executing as a combined module. For lack of specificity, what is recited in the claim cannot make it undeniable clear how this so-recited Integrated Development Environment encompasses in order for it to distinguish from what is used in the rejection, i.e. how more specifically would it perform its manifest file creation that would render it more inventive and different from what the VEA by Koyabashi is achieving. Koyabash's

development tool also puts together beans packages using a integrated compiler having a manifest creator and a visual editor, i.e. Koyabashi's visual builder (VEA) is being treated as an integrated tool or a module, such module having an IDE capability and a beans compiler (with manifest generator). Hence, such visual builder has read on the limitations of (i) and (ii). Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

For the above reasons, the claims will stand rejected as set forth above.

Conclusion

5. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (272) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571)272-3719.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence – please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

VAT
August 8, 2005

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100